Adaptive backward difference formula – discontinuous Galerkin finite element method for the solution of conservation laws

V. Dolejší and P. Kůs

Charles University Prague, Faculty of Mathematics and Physics, Sokolovská 83, 186 75 Praha, Czech Republic, dolejsi@karlin.mff.cuni.cz

Abstract

We deal with the numerical solution of the system of conservation laws. Although this approach has been proposed for a simulation of inviscid compressible flow it can be straightforwardly applied to more general problems. We carried out the space semi-discretization by the discontinuous Galerkin finite element (DGFE) method which is based on a piecewise polynomial discontinuous approximation. The resulting system of ordinary differential equations is discretized by the backward difference formula (BDF). A suitable linearization of the physical fluxes leads to a scheme which is practically unconditionally stable, has a higher order of accuracy with respect to the space and time coordinates and we solve a linear algebraic system at each time level. Moreover, we develop an adaptive technique for a choice of the length of the time step which is based on the use of two BDF of the same order of accuracy. We call the resulting scheme ABDF-DGFE (adaptive BDF-DGFE) method. Finally, the efficiency of the presented adaptive strategy is documented by a set of numerical examples.

Keywords: backward difference formula; discontinuous Galerkin method; adaptive choice of the time step

1 INTRODUCTION

Our aim is to develop a sufficiently efficient, robust and accurate numerical scheme for the solution of a system of time-dependent partial differential equations, particularly for the system of the Euler and Navier-Stokes equations describing a motion of inviscid and viscous compressible flows, respectively. It seems very promising to carry out the space discretization by the *discontinuous Galerkin method* (DGM), which is based on a piecewise polynomial but discontinuous approximation. For a survey of DGM, see [3].

It is possible to use DGM also for the time discretization [21] but the most usual approach is the application of the method of lines. The space semi-discretization of the Euler and particularly of the Navier-Stokes equations leads to a stiff system of ordinary differential equations (ODEs) which should be solved by a suitable method. Many authors employ the explicit Runge-Kutta methods since these schemes have a high order of accuracy and are simple to implement, see, e.g., [1], [2], [18]. Their drawback is a strong restriction of the choice of the time step. In order to avoid this disadvantage, it is suitable to use an implicit time discretization but a full implicit scheme leads to a necessity to solve a nonlinear system of algebraic equations at each time step which is rather expensive. Therefore, we proposed in [11] a semi-implicit time discretization, which is based on a suitable linearization of inviscid fluxes. The linear terms are treated implicitly whereas the nonlinear ones explicitly which leads to a linear algebraic problem at each time step. In [10] we generalized this approach to a higher order approximation with respect to time using a *backward difference formula* (BDF). Then we obtained a semi-implicit scheme which is practically unconditionally stable, has a higher order of accuracy with respect to the space and time coordinates and we solve a linear algebraic system at each time level. We call this scheme the BDF-DGFE (discontinuous Galerkin finite element) method and belongs to the class of the so-called *extrapolated BDF* studied in [4], [22]. Another possibility is to use, e.g., implicit Runge-Kutta methods, which will be the subject of further research.

There is a fundamental question how to choose the length of the time step. Too large time step causes a loss of the accuracy with respect to the time and on the other hand too small time step leads to an inefficiency with respect to the computational time. In order to achieve an optimal choice of the length of the time step with respect to the accuracy and the efficiency a suitable *time step adaptation* technique should be applied. A general time step adaptation strategy is based on a posteriori estimation of a *local discretization error* of the time discretization and using this estimation we choose a time step as large as possible to guarantee that the local discretization error is under a given tolerance.

A usual approach of a posteriori error estimation is the use of two numerical schemes of different orders of accuracy for the solution of a system of ODEs where the numerical solution obtained by the higher order scheme is supposed to be the exact solution and from the difference of both solutions we estimate the local discretization error of the solution obtained by the low order scheme, see, e.g., [16, Sections II.4, III.7]. In [20] a combination of an explicit and an implicit numerical methods of the same order of accuracy was employed. However, we suppose that the use of the explicit scheme can cause some troubles for a large time step due to a loss of the stability. Therefore, we propose in this paper a time step adaptation technique for the BDF-DGFE method, which is based on a combination of two implicit schemes of the same order of accuracy. At each time level we evaluate two numerical solutions using both schemes and from the difference of the solutions we estimate the local discretization error and propose a new time step. We call the resulting scheme the ABDF-DGFE (adaptive BDF-DGFE) method.

We present the ABDF-DGFE technique for the case of the Euler equations which describe inviscid compressible flow. However, this technique (namely the time step adaptation strategy) can be easily extended to different areas of engineering. The contents of the rest of the paper is the following. In section 2 we present a system of Euler equations and mention some properties of the Euler fluxes. In Section 3 we introduce the space semi-discretization of the Euler equation with the aid of the DGM. The main important results of this paper are contained in Section 4, where the time step adaptation strategy is determined for a system of ordinary differential equations. The extension of this approach to the system of the Euler equations is given in Section 5. Four numerical examples demonstrating the efficiency of the time step adaptation technique are presented in Section 6. Finally, we finish with several concluding remarks in Section 7.

2 PROBLEM FORMULATION

A motion of 2D inviscid compressible flow is described by the system of the *Euler equations* written in the conservative form

$$\frac{\partial \boldsymbol{w}}{\partial t} + \sum_{s=1}^{2} \frac{\partial \boldsymbol{f}_s(\boldsymbol{w})}{\partial x_s} = 0 \quad \text{in } Q_T = \Omega \times (0,T), \tag{1}$$

where $\Omega \subset I\!R^2$ is a bounded polygonal domain occupied by gas, $T > 0$ is the length of a time interval,

$$\boldsymbol{w} = (w_1, \dots, w_4)^{\mathrm{T}} = (\rho,\, \rho v_1,\, \rho v_2,\, e)^{\mathrm{T}} \tag{2}$$

is the *state vector* and

$$\boldsymbol{f}_s(\boldsymbol{w}) = (\rho v_s,\, \rho v_s v_1 + \delta_{s1} p,\, \rho v_s v_2 + \delta_{s2} p,\, (e+p)\, v_s)^{\mathrm{T}}, \ s = 1,2, \tag{3}$$

are the *inviscid (Euler) fluxes.* We use the following notation: ρ – density, p – pressure, e – total energy, $\boldsymbol{v} = (v_1, v_2)$ – velocity, δ_{sk} – Kronecker symbol (if $s = k$, then $\delta_{sk} = 1$, else $\delta_{sk} = 0$). The equation of state implies that

$$p = (\gamma - 1)\,(e - \rho |\boldsymbol{v}|^2/2), \tag{4}$$

where $\gamma > 1$ is the Poisson adiabatic constant.

The system (1) is equipped with the initial condition

$$\boldsymbol{w}(x,0) = \boldsymbol{w}^0(x), \quad x \in \Omega, \tag{5}$$

and the boundary conditions

$$B(\boldsymbol{w}) = 0 \quad \text{on } \partial\Omega \times (0,T), \tag{6}$$

chosen in such a way that problem (1) – (6) is linearly well–posed. (See, e.g. [14], Section 3.3.6.) To this end, the boundary $\partial\Omega$ is formed by disjoint parts Γ_{IO} and Γ_W representing the inflow/outflow and impermeable walls, respectively.

On Γ_W we prescribe the impermeability condition

$$\boldsymbol{v} \cdot \boldsymbol{n} = 0 \quad \text{on } \Gamma_W, \tag{7}$$

where $\boldsymbol{n}$ denotes the unit outer normal to $\partial\Omega$. In order to determine boundary conditions on Γ_{IO}, we define the matrix

$$\boldsymbol{P}(\boldsymbol{w}, \boldsymbol{n}) \equiv \sum_{s=1}^{2} \boldsymbol{A}_s(\boldsymbol{w}) n_s, \tag{8}$$

where $\boldsymbol{n} = (n_1, n_2) \in I\!R^2,\ n_1^2 + n_2^2 = 1$ and

$$\boldsymbol{A}_s(\boldsymbol{w}) \equiv \frac{D\boldsymbol{f}_s(\boldsymbol{w})}{D\boldsymbol{w}},\ s = 1, 2, \tag{9}$$

are the Jacobi matrices of the mappings $\boldsymbol{f}_s$. Then we prescribe m_n quantities characterizing the state vector $\boldsymbol{w}$, where m_n is the number of negative eigenvalues of the matrix $\boldsymbol{P}(\boldsymbol{w}, \boldsymbol{n})$ and extrapolate m_p quantities of $\boldsymbol{w}$ from interior of Ω, where $m_p = 4 - m_n$ is the number of non-negative eigenvalues of $\boldsymbol{P}(\boldsymbol{w}, \boldsymbol{n})$. For details, see, e.g., [14].

Using relations (2) – (4), we express the fluxes $\boldsymbol{f}_s,\ s = 1, 2$, in terms of the variables $w_1, \dots, w_4$ in the form

$$\boldsymbol{f}_s(\boldsymbol{w}) = \begin{pmatrix} w_{s+1} \\ \frac{w_{s+1} w_2}{w_1} + \delta_{s1}(\gamma - 1)\left(w_4 - \frac{w_2^2 + w_3^2}{2w_1}\right) \\ \frac{w_{s+1} w_3}{w_1} + \delta_{s2}(\gamma - 1)\left(w_4 - \frac{w_2^2 + w_3^2}{2w_1}\right) \\ \frac{w_{s+1}}{w_1}\left(\gamma w_4 - (\gamma - 1)\frac{w_2^2 + w_3^2}{2w_1}\right) \end{pmatrix}, \quad s = 1, 2. \tag{10}$$

Obviously, $\boldsymbol{f}_s,\ s = 1, 2$, are homogeneous mappings of order one, i.e.,

$$\boldsymbol{f}_s(\omega \boldsymbol{w}) = \omega \boldsymbol{f}_s(\boldsymbol{w}),\ \omega \in I\!R,\ \omega \neq 0,\ i = 1, 2. \tag{11}$$

Then it is easy to show that

$$\boldsymbol{f}_s(\boldsymbol{w}) = \boldsymbol{A}_s(\boldsymbol{w})\boldsymbol{w},\ s = 1, 2. \tag{12}$$

3 SPACE SEMI-DISCRETIZATION

We discretize the system of the Euler equations (1) – (6) with the aid of the *method of lines.* So that within this section we carry out a semi-discretization with respect to the space coordinates by the discontinuous Galerkin method and the resulting system of the ordinary differential equations (ODE) will be discretized by a suitable ODE solver in the next sections.

3.1 Broken Sobolev space

Let $\mathcal{T}_h$ $(h > 0)$ denote a triangulation of the closure $\overline{\Omega}$ of the domain Ω into a finite number of closed elements (triangles or quadrilaterals) K with mutually disjoint interiors. We set $h = \max_{K \in \mathcal{T}_h} \text{diam}(K)$. Let I be a suitable index set such that $\mathcal{T}_h = \{K_i\}_{i \in I}$. If two elements $K_i,\ K_j \in \mathcal{T}_h$ contain a nonempty open part of their faces, we put $\Gamma_{ij} = \Gamma_{ji} = \partial K_i \cap \partial K_j$. For $i \in I$ we set $s(i) = \{j \in I; \Gamma_{ij} \text{ exists}\}$. The boundary $\partial\Omega$ is formed by a finite number of faces of elements K_i adjacent to $\partial\Omega$. We denote all these boundary faces by S_j, where $j \in I_b$ is a suitable index set and put $\gamma(i) = \{j \in I_b; S_j \text{ is a face of } K_i\}$, $\Gamma_{ij} = S_j$ for $K_i \in \mathcal{T}_h$ such that $S_j \subset \partial K_i,\ j \in I_b$. For K_i not containing any boundary face S_j we put $\gamma(i) = \emptyset$. Further we define two disjoint subsets $\gamma_{IO}(i)$ and $\gamma_W(i)$ corresponding to the boundary parts Γ_{IO} and Γ_W, respectively. Obviously, $\gamma(i) = \gamma_{IO}(i) \cup \gamma_W(i)$. Moreover we put $S(i) = s(i) \cup \gamma(i)$ and $\boldsymbol{n}_{ij} = ((n_{ij})_1, (n_{ij})_2)$ is the unit outer normal to ∂K_i on the face Γ_{ij}.

Over the triangulation $\mathcal{T}_h$ we define the *broken Sobolev space*

$$H^k(\Omega, \mathcal{T}_h) = \{v; v|_K \in H^k(K)\ \forall\, K \in \mathcal{T}_h\}, \tag{13}$$

where $H^k(K) = W^{k,2}(K)$ denotes the (classical) Sobolev space on the element K. For $v \in H^1(\Omega, \mathcal{T}_h)$ we set

$$\begin{aligned} v|_{\Gamma_{ij}} &= \text{ trace of } v|_{K_i} \text{ on } \Gamma_{ij}, \\ v|_{\Gamma_{ji}} &= \text{ trace of } v|_{K_j} \text{ on } \Gamma_{ji}, \end{aligned} \tag{14}$$

denoting the *traces* of v on $\Gamma_{ij} = \Gamma_{ji}$, which are different in general. Moreover,

$$[v]_{\Gamma_{ij}} = v|_{\Gamma_{ij}} - v|_{\Gamma_{ji}} \tag{15}$$

denotes the *jump* of function v over the edge Γ_{ij}.

The approximate solution of problem (1) – (6) is sought in the space of discontinuous piecewise polynomial functions $\boldsymbol{S}_h$ defined by

$$\boldsymbol{S}_h \equiv [S_h]^4, \quad S_h \equiv S^p(\Omega, \mathcal{T}_h) \equiv \{v; v|_K \in P^p(K)\ \forall K \in \mathcal{T}_h\}, \tag{16}$$

where p is a positive integer and $P^p(K)$ denotes the space of all polynomials on K of degree at most p. Obviously, $S_h \subset H^1(\Omega, \mathcal{T})$.

3.2 Discontinuous Galerkin method

In order to derive the discrete problem, we multiply (1) by a test function $\boldsymbol{\varphi} \in \left[H^1(\Omega, \mathcal{T}_h)\right]^4$, integrate over any element K_i, $i \in I$, apply Green's theorem and sum over all $i \in I$. In this way we obtain the integral identity

$$\begin{aligned} &\frac{\partial}{\partial t} \sum_{K_i \in \mathcal{T}_h} \int_{K_i} \boldsymbol{w} \cdot \boldsymbol{\varphi}\, \mathrm{d}x \qquad (17)\\ = &\sum_{K_i \in \mathcal{T}_h} \int_{K_i} \sum_{s=1}^{2} \boldsymbol{f}_s(\boldsymbol{w}) \cdot \frac{\partial \boldsymbol{\varphi}}{\partial x_s}\, \mathrm{d}x - \sum_{K_i \in \mathcal{T}_h} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \sum_{s=1}^{2} \boldsymbol{f}_s(\boldsymbol{w}) \cdot \boldsymbol{\varphi}\, (n_{ij})_s\, \mathrm{d}S, \end{aligned}$$

which represents a weak form of the Euler equations in the sense of the broken Sobolev space $H^1(\Omega, \mathcal{T}_h)$ defined by (13).

Now we shall introduce the discrete problem approximating identity (17) with the aid of DGM. To evaluate the boundary integrals in (17) we use the approximation

$$\int_{\Gamma_{ij}} \sum_{s=1}^{2} \boldsymbol{f}_s(\boldsymbol{w}(t))\, (n_{ij})_s \cdot \boldsymbol{\varphi}\, \mathrm{d}S \approx \int_{\Gamma_{ij}} \boldsymbol{H}(\boldsymbol{w}(t)|_{\Gamma_{ij}}, \boldsymbol{w}(t)|_{\Gamma_{ji}}, \boldsymbol{n}_{ij}) \cdot \boldsymbol{\varphi}\, \mathrm{d}S, \tag{18}$$

where $\boldsymbol{H}$ is a *numerical flux*, $\boldsymbol{w}(t)|_{\Gamma_{ij}}$ and $\boldsymbol{w}(t)|_{\Gamma_{ji}}$ are the values of $\boldsymbol{w}$ on Γ_{ij} considered from the interior and the exterior of K_i, respectively, and at time t. For details, see, e.g. [14] or [24].

It is necessary to specify the meaning of $\boldsymbol{w}(t)|_{\Gamma_{ji}}$ for $j \in \gamma(i)$. For inflow/outflow part of the boundary we use non-reflecting boundary conditions in the form

$$\boldsymbol{w}(t)|_{\Gamma_{ji}} \equiv \mathrm{LRP}(\boldsymbol{w}(t)|_{\Gamma_{ij}},\ \boldsymbol{w}_{BC(t)},\ \boldsymbol{n}_{ij}), \quad j \in \gamma_{IO}(i),\ i \in I,\ t \in (0, T), \tag{19}$$

where $\mathrm{LRP}(\cdot,\cdot,\cdot)$ denotes a solution of the *local Riemann problem* considered on edge Γ_{ij}, $j \in \gamma_{IO}(i)$ and $\boldsymbol{w}_{BC}(t)$ is given, e.g., from the far-field boundary conditions. The boundary conditions (19) are perfectly transparent for wave fronts parallel to the boundary, for more details see [10] or [13].

For Γ_{ij}, $j \in \gamma_W(i)$ we use the impermeability condition (7) and replace (18) by the approximation

$$\int_{\Gamma_{ij}} \boldsymbol{H}(\boldsymbol{w}(t)|_{\Gamma_{ij}}, \boldsymbol{w}(t)|_{\Gamma_{ji}}, \boldsymbol{n}_{ij}) \cdot \boldsymbol{\varphi}\, \mathrm{d}S := \int_{\Gamma_{ij}} \boldsymbol{F}_W(\boldsymbol{w}(t), \boldsymbol{n}_{ij}) \cdot \boldsymbol{\varphi}\, \mathrm{d}S, \quad j \in \gamma_W(i), \tag{20}$$

where

$$\boldsymbol{F}_W(\boldsymbol{w}, \boldsymbol{n}) \equiv (0, pn_1, pn_2, 0)^{\mathrm{T}}. \tag{21}$$

The pressure p is expressed in the form

$$p = (\gamma - 1)\, (w_4 - (w_2^2 + w_3^2)/(2w_1)), \tag{22}$$

following from (4) and (2) and extrapolated on Γ_{ij} from K_i and $\boldsymbol{n} = (n_1, n_2) = \boldsymbol{n}_{ij}$.

For $\boldsymbol{w}_h, \boldsymbol{\varphi}_h \in \boldsymbol{S}_h$ we introduce the forms

$$\begin{aligned} (\boldsymbol{w}_h, \boldsymbol{\varphi}_h) &= \int_{\Omega} \boldsymbol{w}_h(x) \cdot \boldsymbol{\varphi}_h(x)\, \mathrm{d}x, \qquad (23)\\ \tilde{\boldsymbol{b}}_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h) &= -\sum_{K_i \in \mathcal{T}_h} \int_{K_i} \sum_{s=1}^{2} \boldsymbol{f}_s(\boldsymbol{w}_h) \cdot \frac{\partial \boldsymbol{\varphi}_h}{\partial x_s}\, \mathrm{d}x \\ &\quad + \sum_{K_i \in \mathcal{T}_h} \sum_{j \in S(i)} \int_{\Gamma_{ij}} \boldsymbol{H}(\boldsymbol{w}_h|_{\Gamma_{ij}}, \boldsymbol{w}_h|_{\Gamma_{ji}}, \boldsymbol{n}_{ij}) \cdot \boldsymbol{\varphi}_h \mathrm{d}S. \end{aligned}$$

Moreover, in order to avoid overshoots and undershoots of the discrete solution near discontinuities (shock waves), we define the jump indicator by

$$g_{K_i}(\boldsymbol{w}_h) \equiv \frac{\sum_{j\in s(i)} \int_{\Gamma_{ij}} [\rho_h]^2 \,\mathrm{d}S}{|K_i|^{3/4} \sum_{j\in s(i)} |\Gamma_{ij}|}, \quad K_i \in^{\mathrm{T}}_h \tag{24}$$

where ρ_h denotes the first component (density) of $\boldsymbol{w}_h \in \boldsymbol{S}_h$, $[\cdot]$ is the inter-element jump defined by (15), $|\Gamma_{ij}|$ is the length of Γ_{ij} and $|K_i|$ is the area of element K_i. This parameter measures the inter-element jumps of the piecewise polynomial function $w_{h,1}$. It vanishes in region where the solution is smooth, see [12]. Moreover, we put

$$g_{\Gamma_{ij}}(\boldsymbol{w}_h) \equiv \frac{1}{2}\left(g_{K_i}(\boldsymbol{w}_h) + g_{K_j}(\boldsymbol{w}_h)\right), \quad j \in s(i),\ i \in I, \tag{25}$$

and define the forms

$$\tilde{\boldsymbol{d}}_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h) \equiv \sum_{i\in I} h_{K_i} g_{K_i}(\boldsymbol{w}_h) \int_{K_i} \nabla \boldsymbol{w}_h \cdot \nabla \boldsymbol{\varphi}_h \,\mathrm{d}x \tag{26}$$

and

$$\tilde{\boldsymbol{J}}_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h) \equiv \sum_{i\in I} \sum_{j\in s(i)} \frac{g_{\Gamma_{ij}}(\boldsymbol{w}_h)}{|\Gamma_{ij}|} \int_{\Gamma_{ij}} [\boldsymbol{w}_h] \cdot [\boldsymbol{\varphi}_h] \,\mathrm{d}S.$$

The first form represents an artificial diffusion and the latter the so-called interior penalty. For more details see [9], [15]. For a simple notation we put

$$\tilde{\boldsymbol{c}}_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h) \equiv \tilde{\boldsymbol{b}}_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h) + \tilde{\boldsymbol{d}}_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h) + \tilde{\boldsymbol{J}}_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h), \quad \boldsymbol{w}_h, \boldsymbol{\varphi}_h \in \boldsymbol{S}_h. \tag{27}$$

Now we can introduce the *semi-discrete problem*.

Definition 1. *Function* $\boldsymbol{w}_h$ *is a* semi-discrete solution *of the problem* (1) – (6)*, if*

$$\begin{aligned} &\text{a)} \quad && \boldsymbol{w}_h \in C^1([0,T]; \boldsymbol{S}_h), \\ &\text{b)} \quad && \left(\frac{\partial \boldsymbol{w}_h(t)}{\partial t}, \boldsymbol{\varphi}_h\right) + \tilde{\boldsymbol{c}}_h(\boldsymbol{w}_h(t), \boldsymbol{\varphi}_h) = 0 \quad \forall \boldsymbol{\varphi}_h \in \boldsymbol{S}_h,\ \forall t \in (0,T), \\ &\text{c)} \quad && \boldsymbol{w}_h(0) = \boldsymbol{w}_h^0, \end{aligned} \tag{28}$$

where $\boldsymbol{w}_h^0 \in S_h$ *denotes an* S_h*-approximation of the initial condition* $\boldsymbol{w}^0$ *from (5).*

Here $C^1([0,T]; \boldsymbol{S}_h)$ is the space of continuously differentiable mappings of the interval $[0,T]$ into $\boldsymbol{S}_h$.

The problem (28), a) – c) constitutes a system of ordinary differential equations for $\boldsymbol{w}_h(t)$ which has to be discretized by a suitable ODE solver. It is possible to use explicit or implicit time discretization. The performance of one explicit time step is usually very fast but the size of the time step is restricted by some type of the *stability condition*, e.g., for the explicit Euler method in the form

$$\tau \le \mathrm{CFL}\, \Lambda(\boldsymbol{w}_h(t)), \quad \Lambda(\boldsymbol{w}_h(t)) \equiv \min_{i\in I} \left(\min_{j\in s(i)} \frac{|K_i|}{r_{ij}(\boldsymbol{w}_h(t), \boldsymbol{n}) |\Gamma_{ij}|} \right), \tag{29}$$

where $r_{ij}(\boldsymbol{w}_h(t), \boldsymbol{n})$ denotes the spectral radius of the matrix $\boldsymbol{P}(\boldsymbol{w}_h(t), \boldsymbol{n})|_{\Gamma_{ij}}$ given by (8) and CFL $\in (0,1)$, see, e.g., [14]. On the other hand, implicit schemes are generally more stable (allow to use a large τ) but en evaluation of each time step requires more computational time. Therefore, the efficiency of the time discretization depends on the balance between the time step restrictions following from the accuracy and stability requirements. For example, shock tube calculations (see [24]), can be done more efficiently with an explicit method than with an implicit one since the accuracy requirement gives $\tau \approx h$ and then the stability condition of type (29) is satisfied naturally.

However, the use of (semi)-implicit time schemes is more efficient in several application, e.g.

- *Low Mach number flow* ($M \ll 1$, $M \equiv |\boldsymbol{v}|/\sqrt{\gamma p/\rho}$ is the Mach number), where the stability condition (29) exhibits a strong restriction since $r_{ij}(\cdot) \approx M^{-1} \gg 1$ and therefore $\Lambda \ll 1$.
- *Steady state flow*, which is solved by the unsteady problem (28), a) – c) for $t \to \infty$. In this case the time step restriction from the accuracy point of view is very weak.

- *Viscous flow*, where the stability condition (29) for explicit schemes should be replaced by

$$\tau \leq \mathrm{CFL} \min \left(\min_{\substack{j \in s(i) \\ i \in I}} \frac{|K_i|}{\lambda_i^{\max} |\Gamma_{ij}|}, \frac{1}{Re} \min_{i \in I} h_{K_i}^2 \right) \tag{30}$$

which is more restrictive for small Reynolds numbers Re than (29).

Our aim is to develop a numerical method for the simulation of viscous compressible flows with a wide range of Mach numbers and various types of flow regimes. Therefore, we employ a higher order time discretization based on the use of the n-step *backward difference formula* (BDF) which belongs to a class of implicit multistep schemes, see [16]. The two-step BDF scheme is A-stable and for $n = 3, 4, 5$ and 6 the methods lose more and more stability. For $n \geq 7$, the formulas are unstable, see [17, Section V.1]. An efficient numerical solution of ODE systems requires a time step adaption technique reflecting a discretization error of the used scheme. So that in the next section we develop an adaptive BDF technique for a general system of ordinary differential equations and the extension of this approach to (28), a) – c) is given in Section 5.

4 TIME DISCRETIZATION

Let us consider a system of ordinary differential equations for an unknown function $y : (0, T) \to I\!R^m$

$$\frac{\mathrm{d}y(t)}{\mathrm{d}t} = F(y), \qquad y(0) = y_0, \tag{31}$$

where $m \in I\!N,\ m \geq 1,\ y_0 \in I\!R^m$ and $F : (0, T) \times I\!R^m \to I\!R^m$ are given. The function $F(\cdot)$ can depend also explicitly on the time t, generally, but since it is not the case of the problem (28), a) – c) we do not consider this case for simplicity.

We assume that (31) has a unique solution. In the following we present two n-step ($n \geq 2$) numerical schemes (the n-step BDF$^{\mathrm{I}}$ and the n-step BDF$^{\mathrm{II}}$) having the same order of accuracy. From a difference of numerical solutions obtained by both methods we derive an estimate of a *local discretization error* and based on this estimate we propose an adaptation strategy of a choice of the size of the time step. This technique we call ABDF (adaptive BDF) method.

4.1 n-step BDF$^{\mathrm{I}}$ and n-step BDF$^{\mathrm{II}}$ schemes

Let $n \geq 2$ be given. We assume that the exact solution of (31) satisfies $y \in [C^{n+2}(0, T)]^m$. By $y^{(i)}(t) \in I\!R^m,\ i = 0, \ldots, n+2$ we denote the i^{th}-derivative of $y(\cdot)$ at $t \in (0, T)$. Let $0 = t_0 < t_1 < t_2 < \ldots t_r = T$ be a partition of the time interval $(0, T)$, we assume that $r \geq n$. We put

$$\begin{aligned} \tau_k &\equiv t_k - t_{k-1}, \quad k = 1, \ldots, r, \\ \theta_k &\equiv \frac{\tau_k}{\tau_{k-1}}, \qquad k = 2, \ldots, r, \end{aligned} \tag{32}$$

denoting the length of the time step and the ratio of lengths of two successive time steps. For a simplicity, we use the notation

$$O(\tau) \equiv O(\tau_k),\ k = 1, \ldots, r. \tag{33}$$

Let y_k denotes an approximate value of the solution $y(t_k)$, i.e.,

$$y_k \approx y(t_k), \qquad k = 0, 1, \ldots, r. \tag{34}$$

Let us assume that $y_{k-l} = y(t_{k-l}),\ l = 1, \ldots, n$ then the *local discretization error* is given by

$$e_k \equiv y(t_k) - y_k, \qquad k = 1, \ldots, r. \tag{35}$$

If the local discretization error of a numerical scheme is equal to $O(\tau^{q+1})$ then we say that the *order of accuracy* of this scheme is equal to q.

We employ two multistep formulas, particularly the n-step BDF$^{\mathrm{I}}$ and the n-step BDF$^{\mathrm{II}}$ schemes given by

$$\sum_{l=0}^{n} \alpha_{n,l}^{\mathrm{I}}\, y_{k-l}^{\mathrm{I}} = \tau_k F(y_k^{\mathrm{I}}), \quad k = n, \ldots, r, \tag{36}$$

	$n=2$	$n=3$
$\alpha^{\mathrm{I}}_{n,0}$	$\frac{2\,\theta_k+1}{\theta_k+1}$	$\frac{\theta_k\theta_{k-1}}{\theta_k\theta_{k-1}+\theta_{k-1}+1}+\frac{2\theta_k+1}{\theta_k+1}$
$\alpha^{\mathrm{I}}_{n,1}$	$-(\theta_k+1)$	$-\frac{(\theta_k+1)(\theta_k\theta_{k-1}+\theta_{k-1}+1)}{\theta_{k-1}+1}$
$\alpha^{\mathrm{I}}_{n,2}$	$\frac{\theta_k^2}{\theta_k+1}$	$\frac{\theta_k^2(\theta_k\theta_{k-1}+\theta_{k-1}+1)}{\theta_k+1}$
$\alpha^{\mathrm{I}}_{n,3}$	—	$-\frac{(\theta_k+1)\theta_k^2\theta_{k-1}^3}{(\theta_{k-1}+1)(\theta_k\theta_{k-1}+\theta_{k-1}+1)}$
c^{I}_n	$-\frac{1+\theta_k}{6\theta_k}$	$-\frac{(\theta_k+1)(\theta_k\theta_{k-1}+\theta_{k-1}+1)}{24\theta_k^2\theta_{k-1}}$

Tab. 1: Values of $\alpha^{\mathrm{I}}_{n,l},\ l=0,\dots,n$ and c^{I}_n for $n=2,3$

and

$$\sum_{l=0}^{n}\alpha^{\mathrm{II}}_{n,l}\,y^{\mathrm{II}}_{k-l}=\frac{\tau_k}{2}\left(F(y^{\mathrm{II}}_k)+F(y^{\mathrm{II}}_{k-1})\right),\qquad k=n,\dots,r, \tag{37}$$

respectively. By y^{I}_{k-l} and y^{II}_{k-l} we denote the approximate solution obtained by the n-step BDF$^{\,\mathrm{I}}$ and the n-step BDF$^{\,\mathrm{II}}$ schemes, respectively. The coefficients $\alpha^{\mathrm{I}}_{n,l},\ l=0,\dots,n$ and $\alpha^{\mathrm{II}}_{n,l},\ l=0,\dots,n$ depend on $\theta_{k-l},\ l=0,\dots,n$, see [16, Section III.5]. The local discretization errors of schemes (36) and (37) are given by

$$e^{\mathrm{I}}_{n,k}\equiv y(t_k)-y^{\mathrm{I}}_k\approx\frac{c^{\mathrm{I}}_n}{\alpha^{\mathrm{I}}_{n,0}}\,\tau_k^{n+1}y^{(n+1)}(t_k)+O(\tau^{n+2})=O(\tau_k^{n+1}),\quad k=n,\dots,r, \tag{38}$$

and

$$\begin{aligned} e^{\mathrm{II}}_{n,k}\equiv y(t_k)-y^{\mathrm{II}}_k &\approx \frac{\mathbf{1}}{\mathbf{2}}\frac{\tau_k^{n+1}}{\alpha^{\mathrm{II}}_{n,0}}\left(c^{\mathrm{I}}_n y^{(n+1)}(t_k)+c^{\mathrm{ex}}_n y^{(n+1)}(t_{k-1})\right)+O(\tau^{n+2}) \\ &= O(\tau_k^{n+1}),\qquad k=n,\dots,r,, \end{aligned} \tag{39}$$

respectively. Moreover, we define the quantity

$$c^{\mathrm{II}}_n\equiv(c^{\mathrm{ex}}_n+c^{\mathrm{I}}_n)/2.$$

The values of coefficients $\alpha^{\mathrm{I}}_{n,l},\alpha^{\mathrm{II}}_{n,l},\ l=0,\dots,n,\ c^{\mathrm{I}}_n,\ c^{\mathrm{ex}}_n$ and c^{II}_n are given in Tables 1 and 2 for $n=2,3$. It is possible two show that the schemes (36) and (37) are stable for $n=2,3$, see [16, Theorem 5.5].

Since the n-step BDF$^{\,\mathrm{I}}$ and the n-step BDF$^{\,\mathrm{II}}$ methods are multi-step, their definitions (36) and (37) can not be used for evaluating of y_k for $k<n$, respectively. The value y_0 is given from the initial condition and the values $y_k,\ k=1,\dots,n-1$ should be evaluated, e.g., by a one-step formula having a sufficient order of accuracy. However, we do not discuss this aspect within this paper.

4.2 Error estimation and time step adaptation

In Section 4.1, we presented two schemes of the same order of accuracy for the numerical solution of problem (31). Within this section we derive an estimation of the local discretization error and based on it, we propose a time step adaptation strategy.

From (38) and (39) we have approximate equalities

$$e^{\mathrm{I}}_{n,k}\equiv\quad y(t_k)-y^{\mathrm{I}}_k\quad\approx\frac{c^{\mathrm{I}}_n}{\alpha^{\mathrm{I}}_{n,0}}\,\tau_k^{n+1}y^{(n+1)}(t_k), \tag{40}$$

$$e^{\mathrm{II}}_{n,k}\equiv\quad y(t_k)-y^{\mathrm{II}}_k\quad\approx\frac{1}{2\,\alpha^{\mathrm{II}}_{n,0}}\,\tau_k^{n+1}\left(c^{\mathrm{I}}_n y^{(n+1)}(t_k)+c^{\mathrm{ex}}_n y^{(n+1)}(t_{k-1})\right). \tag{41}$$

	$n=2$	$n=3$
$\alpha^{\mathrm{II}}_{n,0}$	1	$\frac{\theta_{k-1}(3\theta_k^2+4\theta_k+2)+2(\theta_k+1)}{2(\theta_k+1)(\theta_k\theta_{k-1}+\theta_{k-1}+1)}$
$\alpha^{\mathrm{II}}_{n,1}$	-1	$-\frac{\theta_{k-1}(\theta_k^2+2)+2}{2(\theta_{k-1}+1)}$
$\alpha^{\mathrm{II}}_{n,2}$	0	$+\frac{\theta_k^3\theta_{k-1}}{2(\theta_k+1)}$
$\alpha^{\mathrm{II}}_{n,3}$	—	$-\frac{\theta_k^3\theta_{k-1}^3}{2(\theta_{k-1}+1)(\theta_k\theta_{k-1}+\theta_{k-1}+1)}$
c^{ex}_n	$\frac{\mathbf{1}}{\mathbf{6}\,\boldsymbol{\theta}_{\mathbf{k}}}$	$-\frac{\mathbf{1}+\boldsymbol{\theta}_{\mathbf{k-1}}}{\mathbf{24}\,\boldsymbol{\theta}_{\mathbf{k}}^{\mathbf{2}}\boldsymbol{\theta}_{\mathbf{k-1}}}$
c^{II}_n	$-\frac{\mathbf{1}}{\mathbf{12}}$	$\frac{\mathbf{1}}{\mathbf{2}}\frac{\mathbf{1}+\boldsymbol{\theta}_{\mathbf{k-1}}+(\mathbf{1}+\boldsymbol{\theta}_{\mathbf{k}})(\boldsymbol{\theta}_{\mathbf{k}}\boldsymbol{\theta}_{\mathbf{k-1}}+\boldsymbol{\theta}_{\mathbf{k-1}}+\mathbf{1})}{24\theta_k^2\theta_{k-1}}$

Tab. 2: Values of $\alpha^{\mathrm{II}}_{n,l},\ l=0,\dots,n$ and c^{II}_n for $n=2,3$

Since the local discretization errors are of order $O(\tau^{n+1})$ then the n-step BDF$^{\mathrm{I}}$ and the n-step BDF$^{\mathrm{II}}$ schemes have the order of accuracy equal to n. In order to obtain a computable estimation of the local discretization error we assume that $y^{(n+1)}(t_{k-1}) \approx y^{(n+1)}(t_k)$. Then relation (41) can be written in the form

$$e^{\mathrm{II}}_{n,k} \equiv y(t_k) - y^{\mathrm{II}}_k \approx \frac{c^{\mathrm{II}}_n}{\alpha^{\mathrm{II}}_{n,0}}\,\tau_k^{n+1} y^{(n+1)}(t_k), \tag{42}$$

where c^{II}_n is given by (40).

Subtracting the (approximate) equalities of (40) and (42) we obtain

$$y^{\mathrm{II}}_k - y^{\mathrm{I}}_k \approx \left(\frac{c^{\mathrm{I}}_n}{\alpha^{\mathrm{I}}_{n,0}} - \frac{c^{\mathrm{II}}_n}{\alpha^{\mathrm{II}}_{n,0}}\right)\tau_k^{n+1} y^{(n+1)}(t_k), \tag{43}$$

which implies that

$$y^{(n+1)}(t_k) \approx \frac{y^{\mathrm{II}}_k - y^{\mathrm{I}}_k}{\tau_k^{n+1}}\left(\frac{c^{\mathrm{I}}_n}{\alpha^{\mathrm{I}}_{n,0}} - \frac{c^{\mathrm{II}}_n}{\alpha^{\mathrm{II}}_{n,0}}\right)^{-1}. \tag{44}$$

Finally, from (40), (42) and (44) we obtain the estimation of the local discretization errors for the n-step BDF$^{\mathrm{I}}$ and the n-step BDF$^{\mathrm{II}}$ schemes in the form

$$e^{\mathrm{I}}_{n,k} \approx \delta^{\mathrm{I}}_{n,k}\,(y^{\mathrm{II}}_k - y^{\mathrm{I}}_k), \tag{45}$$

$$e^{\mathrm{II}}_{n,k} \approx \delta^{\mathrm{II}}_{n,k}\,(y^{\mathrm{II}}_k - y^{\mathrm{I}}_k), \tag{46}$$

where

$$\begin{aligned} \delta^{\mathrm{I}}_{n,k} &\equiv \frac{c^{\mathrm{I}}_n}{\alpha^{\mathrm{I}}_{n,0}}\left(\frac{c^{\mathrm{I}}_n}{\alpha^{\mathrm{I}}_{n,0}} - \frac{c^{\mathrm{II}}_n}{\alpha^{\mathrm{II}}_{n,0}}\right)^{-1}, \\ \delta^{\mathrm{II}}_{n,k} &\equiv \frac{c^{\mathrm{II}}_n}{\alpha^{\mathrm{II}}_{n,0}}\left(\frac{c^{\mathrm{I}}_n}{\alpha^{\mathrm{I}}_{n,0}} - \frac{c^{\mathrm{II}}_n}{\alpha^{\mathrm{II}}_{n,0}}\right)^{-1}. \end{aligned} \tag{47}$$

The right-hand-side of (45) and (46) are already computable quantities and they are used for the choice of the time step.

Let $k \geq n$. With the aid of the schemes (36) and (37), we compute the vectors y^{I}_k and y^{II}_k, respectively. As the final approximation of $y_k \approx y(t_k)$ we can put either y^{I}_k or y^{II}_k, but a more efficient approach is to put

$$y_k = \left(\frac{c^{\mathrm{II}}_n}{\alpha^{\mathrm{II}}_{n,0}} - \frac{c^{\mathrm{I}}_n}{\alpha^{\mathrm{I}}_{n,0}}\right)^{-1}\left(y^{\mathrm{I}}_k\frac{c^{\mathrm{II}}_n}{\alpha^{\mathrm{II}}_{n,0}} - y^{\mathrm{II}}_k\frac{c^{\mathrm{I}}_n}{\alpha^{\mathrm{I}}_{n,0}}\right). \tag{48}$$

Then we obtain a scheme having the order of accuracy equal to $n+1$ which follows from a suitable linear combination of (40) and (42) since the terms of order $O(\tau_k^{n+1})$ vanish. Numerical experiments carried out in [19] for a scalar ODE show that the use of (48) gives the order of convergence $n+1$ for $n=2,3$.

Let $e_{n,k}$ denote either $e^{\mathrm{I}}_{n,k}$ or $e^{\mathrm{II}}_{n,k}$. Then (40) and (42) can be written formally by

$$e_{n,k} \approx C\tau_k^{n+1}, \tag{49}$$

where τ_k is the actual time step and C symbolically denotes the rest parts of the right-hand-sides of (40) and (42).

Let $\omega > 0$ be a given error tolerance, i.e., we require that $e_{n,k} \leq \omega$. In virtue of Section 1 it is natural to choose the time step in such a way that

$$e_{n,k} \approx \omega. \tag{50}$$

Based on (49) we deduce that the "optimal" time step $\bar{\tau}_k$ satisfies

$$\omega \approx C\bar{\tau}_k^{n+1}, \tag{51}$$

and from (49) and (51) we obtain

$$\bar{\tau}_k \approx \tau_k \sqrt[n+1]{\frac{\omega}{e_{n,k}}}. \tag{52}$$

So that we define the value

$$\bar{\tau}_k \equiv \tau_k \sqrt[n+1]{\frac{\omega}{\max(e^{\mathrm{I}}_{n,k}, e^{\mathrm{II}}_{n,k})}}, \tag{53}$$

which represents an optimal length of the time step on the k^{th} time level obtained by the error estimates (40) – (41). If this value is not significantly smaller than τ_k then it makes no sense to repeat the evaluation of the solution at the k^{th} time level with the new length of the time step $\bar{\tau}_k$. In this case we use the value $\bar{\tau}_k$ for the length of the next time step τ_{k+1}. Moreover, numerical experiments show that it is not suitable to change the length of the time step very rapidly, so that we include to the proposed adaptive strategy some limitations. Finally, we define the n-step *adaptive backward difference formula* (ABDF) time step algorithm.

n-step ABDF algorithm

```
1) let $\omega > 0$, $n > 0$, $k > 0$ $(k > n)$, $y_{k-l}$, $l = 1, \dots, n$ and $\tau_k > 0$ be given,
2) compute $y^{\mathrm{I}}_k$ and $y^{\mathrm{II}}_k$ by the $n$-step BDF$^{\mathrm{I}}$ (36) and the $n$-step BDF$^{\mathrm{II}}$ (37) schemes, respectively,
3) compute $e^{\mathrm{I}}_{n,k}$ and $e^{\mathrm{II}}_{n,k}$ by (45) and (46), respectively,
4) compute $\bar{\tau}_k$ by (53),
5) *if* $\frac{\bar{\tau}_k}{\tau_k} \geq c_1$
   *then*
      i) put $\tau_{k+1} := \min(\bar{\tau}_k,\ c_2\tau_k)$,
      ii) set $y_k$ using (48),
      iii) put $k := k+1$
      iv) go to step 2)
   *else*
      i) put $\tau_k := \bar{\tau}_k$,
      ii) go to step 2).
```

The constants c_1 and c_2 were derived empirically and they should improve a behaviour of the computational process. We use the values $c_1 = (1.05)^{1/(n+1)}$ (≈ 1.016 for $n=2$ and ≈ 1.012 for $n=3$) and $c_2 = 1.5$.

5 EXTENSION TO THE EULER EQUATIONS

We proceed to the time discretization of the semi-discrete problem (28), a) – c). Since the form $\tilde{\boldsymbol{b}}_h(\cdot,\cdot)$ is nonlinear with respect to its first argument, a direct application of the BDF schemes from the previous section leads to a system of nonlinear algebraic equations at each time step. In order to avoid troubles arising from the nonlinearity, we follow the approach presented in [11] where a semi-implicit discretization of (28), a) – c) was presented. In Section 5.1, we recall the main ideas of the semi-implicit discretization from [11] and in Section 5.2 we apply the time step adaptation technique form the previous Section to the Euler equations. Finally, in Section 5.3, we mention some implementation aspects.

5.1 Linearization

Similarly as in [11], we define a linearization of $\tilde{\boldsymbol{b}}_h(\cdot,\cdot)$. By (23), for $\boldsymbol{w}_h, \boldsymbol{\varphi}_h \in \boldsymbol{S}_h$ we have

$$\begin{aligned}\tilde{\boldsymbol{b}}_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h) &= \underbrace{-\sum_{K_i\in\mathcal{T}_h}\int_{K_i}\sum_{s=1}^{2}\boldsymbol{f}_s(\boldsymbol{w}_h(x))\cdot\frac{\partial\boldsymbol{\varphi}_h(x)}{\partial x_s}\,\mathrm{d}x}_{=:\tilde{\sigma}_1} \qquad (54)\\ &+ \underbrace{\sum_{K_i\in\mathcal{T}_h}\sum_{j\in S(i)}\int_{\Gamma_{ij}}\boldsymbol{H}(\boldsymbol{w}_h|_{\Gamma_{ij}}, \boldsymbol{w}_h|_{\Gamma_{ji}}, \boldsymbol{n}_{ij})\cdot\boldsymbol{\varphi}_h\mathrm{d}S\,.}_{=:\tilde{\sigma}_2}\end{aligned}$$

The individual terms $\tilde{\sigma}_1$ and $\tilde{\sigma}_2$ will be linearized separately. For $\tilde{\sigma}_1$, we employ the property (12) of the Euler fluxes and define the approximation

$$\tilde{\sigma}_1 \approx \sigma_1(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h) \equiv \sum_{K_i\in\mathcal{T}_h}\int_{K_i}\sum_{s=1}^{2}\boldsymbol{A}_s(\bar{\boldsymbol{w}}_h(x))\boldsymbol{w}_h(x)\cdot\frac{\partial\boldsymbol{\varphi}_h(x)}{\partial x_s}\,\mathrm{d}x. \qquad (55)$$

The linearization of the term $\tilde{\sigma}_2$ can be carried out in a simple way, when $\boldsymbol{H}$ in (54) is chosen, for example, as the Vijayasundaram numerical flux, see [23] or [14], Section 3.3.4. The matrix $\boldsymbol{P}(\boldsymbol{w},\boldsymbol{n})$ defined by (8) is diagonalizable: there exist matrices $\boldsymbol{D}$ and $\boldsymbol{T}$ such that

$$\boldsymbol{P}(\boldsymbol{w},\boldsymbol{n}) = \boldsymbol{T}\boldsymbol{D}\boldsymbol{T}^{-1},\ \boldsymbol{D} = \mathrm{diag}\ (\lambda_1,\dots,\lambda_4), \qquad (56)$$

where $\lambda_1,\dots,\lambda_4$ are the eigenvalues of $\boldsymbol{P}$. We define the "positive" and "negative" part of $\boldsymbol{P}$ by

$$\boldsymbol{P}^{\pm}(\boldsymbol{w},\boldsymbol{n}) = \boldsymbol{T}\boldsymbol{D}^{\pm}\boldsymbol{T}^{-1},\ \boldsymbol{D}^{\pm} = \mathrm{diag}\ (\lambda_1^{\pm},\dots,\lambda_4^{\pm}). \qquad (57)$$

Then the Vijayasundaram numerical flux reads

$$\boldsymbol{H}_{VS}(\boldsymbol{w}_1,\boldsymbol{w}_2,\boldsymbol{n}) = \boldsymbol{P}^{+}\left(\frac{\boldsymbol{w}_1+\boldsymbol{w}_2}{2},\boldsymbol{n}\right)\boldsymbol{w}_1 + \boldsymbol{P}^{-}\left(\frac{\boldsymbol{w}_1+\boldsymbol{w}_2}{2},\boldsymbol{n}\right)\boldsymbol{w}_2 \qquad (58)$$

so that we can write

$$\begin{aligned}\tilde{\sigma}_2 &= \sum_{K_i\in\mathcal{T}_h}\sum_{j\in S(i)}\int_{\Gamma_{ij}}\big[\boldsymbol{P}^{+}\left(\langle\boldsymbol{w}_h\rangle_{ij},\boldsymbol{n}_{ij}\right)\boldsymbol{w}_h|_{\Gamma_{ij}} \qquad (59)\\ &\qquad + \boldsymbol{P}^{-}\left(\langle\boldsymbol{w}_h\rangle_{ij},\boldsymbol{n}_{ij}\right)\boldsymbol{w}_h|_{\Gamma_{ji}}\big]\cdot\boldsymbol{\varphi}_h\mathrm{d}S,\end{aligned}$$

where

$$\langle\boldsymbol{w}_h\rangle_{ij} \equiv \frac{1}{2}\left(\boldsymbol{w}_h|_{\Gamma_{ij}} + \boldsymbol{w}_h|_{\Gamma_{ji}}\right). \qquad (60)$$

For $j\in\gamma_W(i)$, in virtue of (20), we use the approximation

$$\boldsymbol{H}(\boldsymbol{w}_h|_{\Gamma_{ij}}, \boldsymbol{w}_h|_{\Gamma_{ji}}, \boldsymbol{n}_{ij})\cdot\boldsymbol{\varphi}_h\mathrm{d}S \approx \int_{\Gamma_{ij}}\boldsymbol{F}_W(\boldsymbol{w}_h,\boldsymbol{n}_{ij})\cdot\boldsymbol{\varphi}\,\mathrm{d}S,\quad j\in\gamma_W(i), \qquad (61)$$

where $\boldsymbol{F}_W$ is given by (21). The vector $\boldsymbol{F}_W$ is a nonlinear function of $\boldsymbol{w}$ and its linearization is given with the aid of the Taylor expansion in the state vector $\bar{\boldsymbol{w}}_h$ as

$$\begin{aligned}\boldsymbol{F}_W(\boldsymbol{w}_h,\boldsymbol{n}) \approx \tilde{\boldsymbol{F}}_W(\bar{\boldsymbol{w}}_h,\boldsymbol{w}_h,\boldsymbol{n}) &\equiv \boldsymbol{F}_W(\bar{\boldsymbol{w}}_h,\boldsymbol{n}) + D\boldsymbol{F}_W(\bar{\boldsymbol{w}}_h,\boldsymbol{n})\left(\boldsymbol{w}_h - \bar{\boldsymbol{w}}_h\right)\\ &= D\boldsymbol{F}_W(\bar{\boldsymbol{w}}_h,\boldsymbol{n})\boldsymbol{w}_h, \qquad (62)\end{aligned}$$

where

$$D\boldsymbol{F}_W(\boldsymbol{w},\boldsymbol{n}) \equiv (\gamma-1)\begin{pmatrix} 0 & 0 & 0 & 0\\ (v_1^2+v_2^2)n_1/2 & -v_1n_1 & -v_2n_1 & n_1\\ (v_1^2+v_2^2)n_2/2 & -v_1n_2 & -v_2n_2 & n_2\\ 0 & 0 & 0 & 0\end{pmatrix} \qquad (63)$$

is obtained by the differentiation of function $\boldsymbol{F}_W$ given by (21) with respect to $\boldsymbol{w} = (w_1,\dots,w_4)$. Here $\boldsymbol{n} = (n_1,n_2),\ v_j = w_{j+1}/w_1,\ j = 1,2.$

Finally, the form (58) and relations (61) – (62) offer the linearized approximation

$$\begin{aligned} & \sigma_2(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h) \\ \equiv & \sum_{K_i \in \mathcal{T}_h} \sum_{j \in \gamma_W(i)} \int_{\Gamma_{ij}} \tilde{\boldsymbol{F}}_W(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{n}_{ij}) \cdot \boldsymbol{\varphi} \, \mathrm{d}S, \\ & + \sum_{K_i \in \mathcal{T}_h} \sum_{j \in \gamma_{IO}(i)} \int_{\Gamma_{ij}} \left[\boldsymbol{P}^+ \left(\langle \bar{\boldsymbol{w}}_h \rangle_{ij}, \boldsymbol{n}_{ij} \right) \bar{\boldsymbol{w}}_h|_{\Gamma_{ij}} + \boldsymbol{P}^- \left(\langle \bar{\boldsymbol{w}}_h \rangle_{ij}, \boldsymbol{n}_{ij} \right) \bar{\boldsymbol{w}}_h|_{\Gamma_{ji}} \right] \cdot \boldsymbol{\varphi}_h \mathrm{d}S \\ & + \sum_{K_i \in \mathcal{T}_h} \sum_{j \in s(i)} \int_{\Gamma_{ij}} \left[\boldsymbol{P}^+ \left(\langle \bar{\boldsymbol{w}}_h \rangle_{ij}, \boldsymbol{n}_{ij} \right) \boldsymbol{w}_h|_{\Gamma_{ij}} + \boldsymbol{P}^- \left(\langle \bar{\boldsymbol{w}}_h \rangle_{ij}, \boldsymbol{n}_{ij} \right) \boldsymbol{w}_h|_{\Gamma_{ji}} \right] \cdot \boldsymbol{\varphi}_h \mathrm{d}S, \end{aligned} \tag{64}$$

Finally, we define the form

$$\boldsymbol{b}_h(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h) \equiv -\sigma_1(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h) + \sigma_2(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h), \tag{65}$$

where σ_1 and σ_2 are given by (55) and (64), respectively. The form $\boldsymbol{b}_h$ is linear with respect to the second and third variable and from (12), (55), (59), (61) and (62) it follows that the form $\boldsymbol{b}_h(\cdot,\cdot,\cdot)$ is consistent with $\tilde{\boldsymbol{b}}_h(\cdot,\cdot)$ in the following way

$$\boldsymbol{b}_h(\boldsymbol{w}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h) = \tilde{\boldsymbol{b}}_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h) \quad \forall \boldsymbol{w}_h, \boldsymbol{\varphi}_h \in \boldsymbol{S}_h. \tag{66}$$

In a natural way, we define the linearized analogues of forms (26) and (27) by

$$\boldsymbol{d}_h(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h) \equiv \sum_{i \in I} h_{K_i} g_{K_i}(\bar{\boldsymbol{w}}_h) \int_{K_i} \nabla \boldsymbol{w}_h \cdot \nabla \boldsymbol{\varphi}_h \, \mathrm{d}x \tag{67}$$

and

$$\boldsymbol{J}_h(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h) \equiv \sum_{i \in I} \sum_{j \in s(i)} \frac{g_{\Gamma_{ij}}(\bar{\boldsymbol{w}}_h)}{|\Gamma_{ij}|} \int_{\Gamma_{ij}} [\boldsymbol{w}_h] \cdot [\boldsymbol{\varphi}_h] \, \mathrm{d}S,$$

respectively. Obviously, forms $\boldsymbol{d}_h(\cdot,\cdot,\cdot)$ and $\boldsymbol{J}_h(\cdot,\cdot,\cdot)$ are consistent with forms $\tilde{\boldsymbol{d}}_h(\cdot,\cdot)$ and $\tilde{\boldsymbol{J}}_h(\cdot,\cdot)$ by similar relations as (66), respectively. Finally, for a shorter notation we put

$$\boldsymbol{c}_h(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h) \equiv \boldsymbol{b}_h(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h) + \boldsymbol{d}_h(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h) + \boldsymbol{J}_h(\bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h), \quad \bar{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h \in \boldsymbol{S}_h. \tag{68}$$

5.2 Full space-time discretization

The main idea of the semi-implicit discretization is to threat the linear part of $\boldsymbol{c}_h$ (represented by its second argument) implicitly and the nonlinear part of $\boldsymbol{c}_h$ (represented by its first argument) explicitly. Then for the linear part of $\boldsymbol{c}_h(\cdot,\cdot,\cdot)$ we employ the n-step BDF introduced in Section 4 and for the nonlinear part of $\boldsymbol{c}_h(\cdot,\cdot,\cdot)$ we employ a suitable explicit higher order extrapolation which preserve a given order of accuracy and does not destroy the linearity of the algebraic problem at each time level.

Let $n \geq 2$, $0 = t_0 < t_1 < t_2 < \dots t_r = T$ be a partition of the time interval $(0,T)$ and $\boldsymbol{w}_h^k \in \boldsymbol{S}_h$ denotes a piecewise polynomial approximation of $\boldsymbol{w}_h(t_k)$, $k = 0, 1, \dots, r$. In order to employ an explicit extrapolation in the first argument of $\boldsymbol{c}_h$ at the time level t_k, $k \geq n$, we define the Lagrangian interpolation polynomial $\tilde{\boldsymbol{w}} : (0,T) \to \boldsymbol{S}_h$ through the pairs $(t_{k-l}, \boldsymbol{w}_h^{k-l}) \in (I\!R, \boldsymbol{S}_h)$, $l = 1, \dots, n$ by

$$\tilde{\boldsymbol{w}}(t) \equiv \sum_{l=1}^{n} \boldsymbol{w}_h^{k-l} \tilde{\ell}_{n,k-l}(t), \tag{69}$$

where

$$\tilde{\ell}_{n,k-l}(t) \equiv \prod_{\substack{j=1 \\ j \neq l}}^{n} (t - t_{k-j}) \left(\prod_{\substack{j=1 \\ j \neq l}}^{n} (t_{k-l} - t_{k-j}) \right)^{-1}. \tag{70}$$

Obviously, $\tilde{\boldsymbol{w}}(t_{k-l}) = \boldsymbol{w}_h^{k-l}$, $l = 1, \dots, n$. Now, for the first argument of the form $\boldsymbol{c}_h$ we use the evaluation of the Lagrangian polynomial $\tilde{\boldsymbol{w}}(t)$ at $t := t_k$ given by

$$\tilde{\boldsymbol{w}}(t_k) = \sum_{l=1}^{n} \beta_{n,l} \boldsymbol{w}_h^{k-l}, \tag{71}$$

	$n=2$	$n=3$
$\beta_{n,1}$	$1+\theta_k$	$(1+\theta_k)\frac{\theta_k\theta_{k-1}+\theta_{k-1}+1}{\theta_{k-1}+1}$
$\beta_{n,2}$	$-\theta_k$	$-\theta_k(\theta_k\theta_{k-1}+\theta_{k-1}+1)$
$\beta_{n,3}$	—	$\theta_k\theta_{k-1}\frac{\theta_k\theta_{k-1}+\theta_{k-1}}{\theta_{k-1}+1}$

Tab. 3: Values of coefficients $\beta_{n,l},\ l=1,\dots,n$ for $n=2,3$

where

$$\beta_{n,l} \equiv \tilde{\ell}_{n,k-l}(t_k), \quad l=1,\dots,n. \tag{72}$$

The relations for $\beta_{n,l},\ l=1,\dots,n,\ n=2,3$ are given in Table 3.

Finally, we arrive at the definition of two semi-implicit n-step BDF-DGFE schemes. We start from the semi-discrete DGFE problem (28), a) – c), where the time derivative term is approximated either by the n-step BDF$^{\mathrm{I}}$ or by the n-step BDF$^{\mathrm{II}}$ formulas, the form $\tilde{\boldsymbol{c}}_h(\cdot,\cdot)$ is replaced by the form $\boldsymbol{c}_h(\cdot,\cdot,\cdot)$ whose second argument is considered on the new time level and for the first one, the higher order explicit extrapolation of type (71) is employed.

Definition 2. *Let $n\geq 2$, we define the* approximate solution *of problem (1) – (6) by the n-step BDF$^{\mathrm{I}}$-DGFE scheme as functions $\boldsymbol{w}_{h,k},\ k=1,\dots,r$, satisfying the conditions*

$$\begin{aligned}
&a) && \boldsymbol{w}_{h,k}\in\boldsymbol{S}_h,\\
&b) && \frac{1}{\tau_k}\left(\sum_{l=0}^{n}\left(\alpha^{\mathrm{I}}_{n,l}\boldsymbol{w}_{h,k-l}\right),\ \boldsymbol{\varphi}_h\right)+\boldsymbol{c}_h\left(\sum_{l=1}^{n}\left(\beta_{n,l}\boldsymbol{w}_{h,k-l}\right),\ \boldsymbol{w}_{h,k},\ \boldsymbol{\varphi}_h\right)=0\\
& && \qquad\qquad\qquad\forall\,\boldsymbol{\varphi}_h\in\boldsymbol{S}_h,\ k=n-1,\dots,r-1,\\
&c) && \boldsymbol{w}_{h,0}\in S_h \text{ is an approximation of } \boldsymbol{w}^0,\\
&d) && \boldsymbol{w}_{h,l}\in\boldsymbol{S}_h,\ l=1,\dots,n-1 \text{ are given by a suitable one-step method,}
\end{aligned}\tag{73}$$

where the coefficients $\alpha^{\mathrm{I}}_{n,l},\ l=0,\dots,n$ and $\beta_{n,l},\ l=1,\dots,n$ given by (36) and (72), respectively.

Definition 3. *Let $n\geq 2$, we define the* approximate solution *of problem (1) – (6) by the n-step BDF$^{\mathrm{II}}$-DGFE scheme as functions $\boldsymbol{w}_{h,k},\ k=1,\dots,r$, satisfying the conditions*

$$\begin{aligned}
&a) && \boldsymbol{w}_{h,k}\in\boldsymbol{S}_h,\\
&b) && \frac{1}{\tau_k}\left(\sum_{l=0}^{n}\left(\alpha^{\mathrm{II}}_{n,l}\boldsymbol{w}_{h,k-l}\right),\ \boldsymbol{\varphi}_h\right)+\frac{1}{2}\boldsymbol{c}_h\left(\sum_{l=1}^{n}\left(\beta_{n,l}\boldsymbol{w}_{h,k-l}\right),\ \boldsymbol{w}_{h,k},\ \boldsymbol{\varphi}_h\right)\\
& && \qquad\qquad=-\frac{1}{2}\boldsymbol{c}_h\left(\boldsymbol{w}_{h,k-1},\ \boldsymbol{w}_{h,k-1},\ \boldsymbol{\varphi}_h\right)\\
& && \qquad\qquad\qquad\forall\,\boldsymbol{\varphi}_h\in\boldsymbol{S}_h,\ k=n-1,\dots,r-1,\\
&c) && \boldsymbol{w}_{h,0}\in S_h \text{ is an approximation of } \boldsymbol{w}^0,\\
&d) && \boldsymbol{w}_{h,l}\in\boldsymbol{S}_h,\ l=1,\dots,n-1 \text{ are given by a suitable one-step method,}
\end{aligned}\tag{74}$$

where the coefficients $\alpha^{\mathrm{II}}_{n,l},\ l=0,\dots,n$ and $\beta_{n,l},\ l=1,\dots,n$ given by (37) and (72), respectively.

The problems (73), a) – d) and (74), a) – d) represent systems of linear algebraic equations for each $k=n-1,\dots,r-1$ which should be solved by a suitable solver, see Section 5.3.

Following the approach from Section 4.2, we introduce the adaptive BDF - DGFE algorithm for the system of the Euler equations. Based on (45) – (46), we define estimations of the local discretization errors of the n-step BDF$^{\mathrm{I}}$-DGFE and the n-step BDF$^{\mathrm{II}}$-DGFE schemes by

$$e^{\mathrm{I}}_{n,k} \approx \delta^{\mathrm{I}}_{n,k}\,\|\boldsymbol{w}^{\mathrm{II}}_{h,k}-\boldsymbol{w}^{\mathrm{I}}_{h,k}\|_{L^2(\Omega)}, \tag{75}$$

$$e^{\mathrm{II}}_{n,k} \approx \delta^{\mathrm{II}}_{n,k}\,\|\boldsymbol{w}^{\mathrm{II}}_{h,k}-\boldsymbol{w}^{\mathrm{I}}_{h,k}\|_{L^2(\Omega)}, \tag{76}$$

respectively, where $\delta^{\mathrm{I}}_{n,k}$ and $\delta^{\mathrm{II}}_{n,k}$ are given by (47) and $\boldsymbol{w}^{\mathrm{I}}_{h,k}$ and $\boldsymbol{w}^{\mathrm{II}}_{h,k}$ are given by (73) and (74), respectively. Moreover, based on (53) we define the optimal value of the length of the time step by

$$\bar{\tau}_k \equiv \tau_k \sqrt[n+1]{\frac{\omega}{\max(e^{\mathrm{I}}_{n,k}, e^{\mathrm{II}}_{n,k})}}, \tag{77}$$

where $\omega > 0$ is a given tolerance and $e^{\mathrm{I}}_{n,k}$ and $e^{\mathrm{II}}_{n,k}$ are given by (75) and (76), respectively. For the final approximation of $\boldsymbol{w}_{h,k} \approx \boldsymbol{w}_h(t_k)$ it is possible to use a relation similar to (48), but a number of numerical experiments shows that this choice leads to unsatisfactory results. Therefore, we simply put $\boldsymbol{w}_{h,k} := \boldsymbol{w}^{\mathrm{I}}_{h,k}$.

Finally, we define the n**-step ABDF-DGFE** algorithm for the system of the Euler equations in the same way as the n-step ABDF algorithm for ODE introduced at the end of Section 4.2 with the following modifications

1. we replace y^{I}_k and y^{II}_k by $\boldsymbol{w}^{\mathrm{I}}_{h,k}$ and $\boldsymbol{w}^{\mathrm{II}}_{h,k}$ given by the n-step BDF$^{\mathrm{I}}$ (73) and the n-step BDF$^{\mathrm{II}}$ (74) schemes, respectively,
2. the estimates of the local discretization errors $e^{\mathrm{I}}_{n,k}$ and $e^{\mathrm{II}}_{n,k}$ are computed by (75) and (76), respectively,
3. if the time step is successful we put $\boldsymbol{w}_{h,k} := \boldsymbol{w}^{\mathrm{I}}_{h,k}$ in the step 5) ii).

Remark 1. *Based on numerical experiments carried out for a scalar nonlinear convection-diffusion equation in [8] we suppose that the orders of convergence of schemes (73), a) – d) and (74), a) – d) are*

$$O(h^p + \tau^n) \tag{78}$$

(in $L^\infty((0,T);H^1(\Omega))$), where p is the degree of polynomial approximation with respect to space and n is the degree of the multi-step BDF.

5.3 Several implementation remarks

Within this section we mention some aspects of the implementation of the n-step ABDF-DGFE method to the Euler equations.

Linear algebraic system solver As we mentioned in the previous Section, problems (73), a)–d) and (74), a)–d) represent systems of linear algebraic equations. Let $\{\boldsymbol{\psi}_l\}_{l=0}^{\mathrm{dof}}$ represent a basis of the space of vector-valued discontinuous piecewise polynomial functions $\boldsymbol{S}_h$ defined by (16), where $\mathrm{dof}(= 2\,(p+1)(p+2)\,\#\mathcal{T}_h)$ denotes the dimension of $\boldsymbol{S}_h$. Then a function $\boldsymbol{w}_{h,k} \in \boldsymbol{S}_h$ can be written in the form

$$\boldsymbol{w}_{h,k}(x) = \sum_{l=1}^{\mathrm{dof}} \xi_{k,l}\boldsymbol{\psi}_l(x), \quad x \in \Omega,\ k = 0,1,\dots,r, \tag{79}$$

where $\xi_{k,l} \in I\!R,\ l = 1,\dots,\mathrm{dof},\ k = 0,\dots,r$. Moreover, for $\boldsymbol{w}_{h,k} \in \boldsymbol{S}_h$ we define a vector of its basis coefficients by

$$\boldsymbol{W}_k \equiv (\xi_{k,1},\ \xi_{k,2},\dots,\xi_{k,\mathrm{dof}}) \in I\!R^{\mathrm{dof}}, \quad k = 0,1,\dots,r. \tag{80}$$

Then the linear algebraic problems (73) and (74) can be written in the matrix form

$$\left(\alpha^{\mathrm{I}}_{n,0}\boldsymbol{M} + \tau_k\boldsymbol{C}_k\right)\boldsymbol{W}_k = \boldsymbol{q}^{\mathrm{I}}_k, \quad k = n,\dots,r, \tag{81}$$

and

$$\left(\alpha^{\mathrm{II}}_{n,0}\boldsymbol{M} + \frac{\tau_k}{2}\boldsymbol{C}_k\right)\boldsymbol{W}_k = \boldsymbol{q}^{\mathrm{II}}_k, \quad k = n,\dots,r, \tag{82}$$

respectively, where matrix $\boldsymbol{M}$ is the *mass matrix* given by

$$\boldsymbol{M} = \{\boldsymbol{M}_{ij}\}_{i,j=1}^{\mathrm{dof}}, \quad \boldsymbol{M}_{ij} \equiv \int_\Omega \boldsymbol{\psi}_i \cdot \boldsymbol{\psi}_j \,\mathrm{d}x,\ i,j = 1,\dots,\mathrm{dof}, \tag{83}$$

$\boldsymbol{C}_k$ is a the matrix corresponding to form $\boldsymbol{c}(\cdot,\cdot,\cdot)$ defined by

$$\boldsymbol{C}_k = \{\boldsymbol{C}_{k,ij}\}_{i,j=1}^{\mathrm{dof}}, \quad \boldsymbol{C}_{k,ij} \equiv \boldsymbol{c}_h\left(\sum_{l=1}^{n}\beta_{n,l}\boldsymbol{w}_{h,k-l},\ \boldsymbol{\psi}_i,\ \boldsymbol{\psi}_j\right),\ i,j = 1,\dots,\mathrm{dof}, \tag{84}$$

and $\boldsymbol{q}_k^{\rm I},\ \boldsymbol{q}_k^{\rm II} \in I\!R^{\rm dof}$ represent the right-hand-sides of (73) and (74) given by

$$\boldsymbol{q}_k^{\rm I} = \{\boldsymbol{q}_k^{\rm I}\}_{i=1}^{\rm dof}, \quad \{\boldsymbol{q}_k^{\rm I}\}_i \equiv -\left(\sum_{l=1}^{n} \alpha_{n,l}^{\rm I} \boldsymbol{w}_{h,k-l},\ \boldsymbol{\psi}_i\right), \tag{85}$$

$$\boldsymbol{q}_k^{\rm II} = \{\boldsymbol{q}_k^{\rm II}\}_{i=1}^{\rm dof}, \quad \{\boldsymbol{q}_k^{\rm II}\}_i \equiv -\left(\sum_{l=1}^{n} \alpha_{n,l}^{\rm II} \boldsymbol{w}_{h,k-l},\ \boldsymbol{\psi}_i\right) - \frac{\tau_k}{2} \boldsymbol{c}_h\left(\boldsymbol{w}_{h,k-1},\ \boldsymbol{w}_{h,k-1},\ \boldsymbol{\psi}_i\right).$$

The linear algebraic problems (81) and (82) should be numerically solved at each time level $t_k,\ k = n, \dots, r$. It is possible to use a direct solver which is more efficient for not too large dof (usually dof $\approx 10^4 - 10^5$). For larger systems it is suitable to use some iterative solvers, e.g., GMRES with a preconditioning. Numerical experiments show that the use of two n-step BDF-DGFE schemes at each time step does not cause any essential increase of the CPU-time in comparison with the use of only one scheme. It is caused by the fact that the elements of matrix $\boldsymbol{C}_k$ should be evaluated only once for each k and the CPU-time for the setting of matrices on the left-hand-sides of (81) – (82) is almost negligible when compared to the evaluation of the elements of $\boldsymbol{C}_k$. Moreover, the approximate solution of (81) $\boldsymbol{W}_k^{\rm I}$ can be used as an initial approximation of the solution $\boldsymbol{W}_k^{\rm II}$ of the second problem (82) in an iterative process. Since the vectors $\boldsymbol{W}_k^{\rm I}$ and $\boldsymbol{W}_k^{\rm II}$ are close to each other (both represents approximation of $\boldsymbol{w}(t_k)$), the iterative process for the second problem is very fast in comparison with the solution of the first problem.

Start of the computational process We should still precise the computation of the first $n-1$ functions $\boldsymbol{w}_{h,k},\ k = 1, \dots, n-1$ followed from (73), d) and (74), d). We use the one-step backward Euler formula for the computation of $\boldsymbol{w}_{h,1}$ and then the m-step ABDF-DGFE scheme for $\boldsymbol{w}_{h,m},\ m = 2, \dots, n-1$. This strategy causes some loss of the accuracy in the first $n-1$ steps of course, it would be useful to apply some one-step higher order formula (e.g. the Runge-Kutta scheme) but we do not implement it yet. We suppose that the loss of the accuracy is not significant in industrial applications.

Concerning the choice of the length of the first step τ_1 we use the stability condition (29) derived for an explicit finite volume scheme (see [14]) with CFL = 1. Furthermore, we put $\tau_2 := \tau_1$ and the ABDF-DGFE strategy is applied from the second time step.

6 NUMERICAL EXAMPLES

In this section we present several numerical examples which demonstrate the efficiency of the proposed time step adaptation technique. By the efficiency we mean the number of time steps necessary to achieve the final time $t = T$. Of course, from the practical point of view a measure of the efficiency is the computational (CPU) time, but there are other computational aspects. E.g., a CPU-time of a solution of the linear algebraic problems (73), b) and (74), b) by a *direct solver* is almost independent of the length of the time step whereas a CPU-time of a solution of the linear problem by an iterative solver strongly increases with the length of the time step. It is caused by the fact that for a small time step $\tau_k,\ k > 0$ the numerical solutions at the time levels t_{k-1} and t_k are close to each other and when $\boldsymbol{w}_{h,k-1}$ is used as the initial approximation for computing $\boldsymbol{w}_{h,k}$ only a few iterations of the iterative solver should be evaluated. However, the use of a suitable preconditioning decreases very much the CPU-time. A development of an efficient scheme for the solution of the linear problem is not a subject of this paper. So that we consider the efficiency of the schemes in the terms of the number of time steps.

We start with a model scalar ODE and demonstrate the efficiency of the n-step ABDF ($n = 2, 3$) algorithm in comparison with the BDF scheme with a fixed length of the time step. Further, we deal with a simulation of steady inviscid compressible flow. We compare the efficiency of the n-step ABDF-BDF scheme with a computation with an almost fixed time step and a scheme with an empirically increasing length of the time step from [11] for achieving the steady state solution. Finally, two examples of unsteady compressible flow are presented. The first one exhibits a low Mach number flow through the GAMM channel with a periodical generation of pressure pulses at the outflow part of the channel. The second represents a supersonic flow through the forward facing step.

6.1 Scalar ordinary differential equation

Let us consider the following model ordinary differential equation

$$y' = \frac{\alpha e^{\alpha t}}{e^{\alpha} - 1} \tag{86}$$

prescribed tolerance for global error		10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
n-step BDF	$n=2$	642	1425	3197	6972	15110
τ_k =const.	$n=3$	410	855	1586	2879	5222
n-step ABDF	$n=2$	26	36	65	145	266
	$n=3$	24	29	43	70	108

Tab. 4: Number of time steps necessary to achieve the prescribed tolerance for the global error from 10^{-2} till 10^{-6} at $t=1$ for (86) for fixed and adaptively chosen time steps

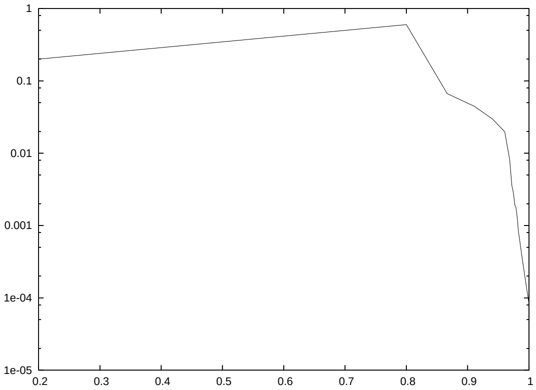


Fig. 1: The length of the time step in the logarithmic scale for the 3-step ABDF scheme for $t \in (0,1)$ and the prescribed global error 10^{-6}

for $y:(0,1) \to I\!R$ with the initial condition $y(0)=0$ and the parameter $\alpha=500$. The analytical solution of (86) has the form

$$y = \frac{e^{\alpha t}-1}{e^{\alpha}-1}. \tag{87}$$

Since the exact solution is known we can compute the global computational error ($g_k \equiv y(t_k) - y_k$) exactly. We carried out several computations with the aid of the n-step ABDF scheme ($n=2,3$) defined at the end of Section 4.2 and with the n-step BDF scheme with a fixed time step. We chose the tolerance ω for the first scheme and the length of the time step for the second scheme in such a way that the (computable) global errors at $t=1$ are equal to 10^{-2}, 10^{-3}, 10^{-4}, 10^{-5} and 10^{-6} (we admit a relative difference of 1%). Table 4 contains the number of the time steps necessary to achieve the prescribed errors at $t=1$. We observe that the number of time steps is significantly smaller for the adaptive scheme. It is caused by the fact that the exact solution of (86) exponentially grows up and then it is possible to start with a large length of the time step and then to reduce it for increasing t. This effect is documented in Figure 1, where the length of the time step for the 3-step ABDF scheme for $t \in (0,1)$ is shown.

6.2 Steady state transonic flow

We solve the steady state inviscid compressible flow through the well-known benchmark GAMM channel, see Figure 2, left, where the computational domain with a triangular grid having 1417 elements is shown. This grid was obtained by an anisotropic mesh adaptation strategy from [5], [6]. We carried out several computations with the inlet Mach number $M_{\rm in}=0.67$ which produces a transonic flow with a strong discontinuity (shock wave), see Figure 2, right. The piecewise linear approximation was employed for all results presented in Section 6.2.

We seek the steady state solution by the time stabilization approach where the computational process

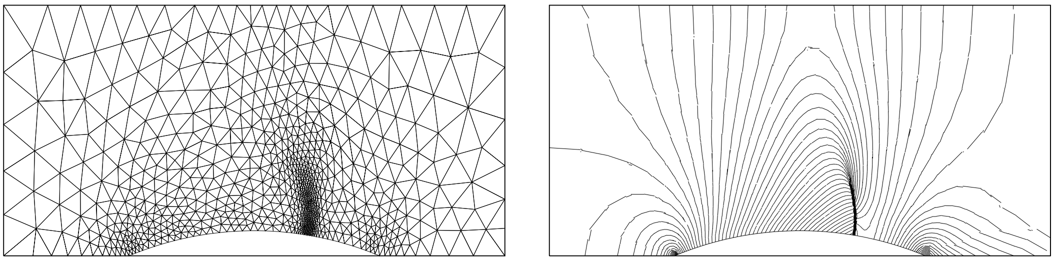


Fig. 2: GAMM channel, used adapted triangulation (left) and steady-state solution – isolines of Mach number (right)

is carried out for “$t \to \infty$”. As a stopping criterion we employ the condition

$$\mathrm{res}(k) \equiv \frac{1}{\tau_k} \frac{\|\rho_{h,k} - \rho_{h,k-1}\|_{L^2(\Omega)}}{\|\rho_{h,k}\|_{L^2(\Omega)}} \leq \varepsilon, \tag{88}$$

where $\rho_{h,k}$ denotes a discontinuous piecewise polynomial approximation of the density (first component of $\boldsymbol{w}_{h,k}$) at the time level t_k, τ_k is the length of the time step and $\varepsilon > 0$ is a prescribed tolerance. In the computations presented within this section we put $\varepsilon = 10^{-5}$.

We carried out three types of the choice of the length of the time step

A1 2-step ABDF-DGFE scheme presented in Section 5.2,

A2 2-step BDF-DGFE scheme, where the length of the time step is given by the stability condition (29) with fixed CFL number,

A3 2-step BDF-DGFE scheme with an increasing CFL number, where the time step is given by (29), where the value CFL is increased according to the formula

$$\mathrm{CFL} := \mathrm{CFL}(t_k) \equiv \mathrm{CFL}_{\max} - (\mathrm{CFL}_{\max} - 1)\exp(-\eta\, t_k), \tag{89}$$

where $\mathrm{CFL}_{\max}$ is the upper limit of the CFL number, t_k is the actual time and $\eta > 0$ is a constant, we put $\eta = 0.1$, see [11].

There is of course a question how to choose the tolerance ω for approach A1, the CFL number for approach A2 and the value $\mathrm{CFL}_{\max}$ for approach A3. Although the n-step BDF schemes are sufficiently stable for $n = 2, 3$, numerical experiments show that too long time step can cause a failure of the computational process due to (usually) unphysical initial conditions. Therefore, we carried out a lot of numerical experiments and found the values of parameters which allow the largest admissible lengths of the time steps for A1 – A3. Table 5 shows the number of time steps necessary to achieve the steady state solution for the approaches A1 – A3 with the values of parameters used in the computations. Figure 3, left shows the history of the convergence to the steady-state solution for A1 – A3, i.e. the dependence of the residuum res(k) given by (88) on k. Moreover, Figure 3, right shows the corresponding dependence of the lengths of the time steps on t. We observe that the time step is almost constant for approach A2, it grows for A3 whereas it changes adaptively for A1.

Since we seek the steady state solution it is possible to use the one step BDF (implicit Euler scheme) of course. Although the one step method is much simple for implementation it does not exhibit any essential save of the computational time. It is caused by the fact that more than 99 % of the computational time is needed for the setting of matrix $\boldsymbol{C}_k$ given by (84) and for the solution of the algebraic linear problems (81) – (82). These computational operations do not depend of the degree of the employed multistep formula.

6.3 Unsteady low Mach number flow

The third example exhibits unsteady low Mach number flow through the GAMM channel. The initial condition is the steady state solution for the inflow Mach number $M = 10^{-3}$. Figure 4, left shows the relatively coarse triangular grid used, having 1012 elements and Figure 4, right shows the isolines of pressure of the steady state solution, which was obtained by the piecewise cubic approximation with respect to the space. We use the non-reflecting boundary conditions (19) with

$$\boldsymbol{w}_{BC} = (1, 1, 0, 1\,785\,714.786), \tag{90}$$

case	method	parameters	#iterations
A1	2-step ABDF-DGFE	$\eta = 0.02$	1672
A2	2-step BDF-DGFE, CFL fixed	CFL $= 20$	2723
A3	2-step BDF-DGFE, CFL variable	$\mathrm{CFL}_{\max} = 50$	2277

Tab. 5: Comparison of the number of used time step for the computations A1 – A3 necessary in order to achieve the steady state solution

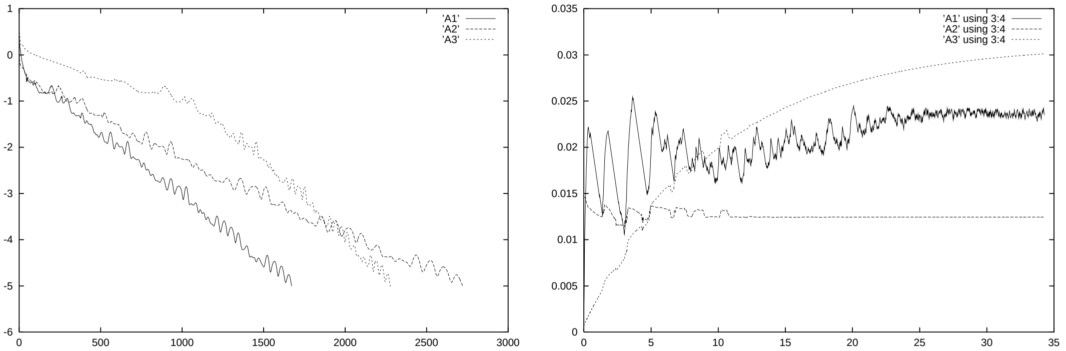


Fig. 3: GAMM channel, steady-state solution, the dependence of res(k) on k (left) and the dependence of the size of the time step on t (right) for approaches A1 – A3

which (using (4)) gives the inflow/output pressure $p_0 = 714\,285.7143$.

At $t = 0$ we start to periodically modify the pressure at outflow of the channel according to the formula

$$p(t) = p_0(1 + \bar{p}(t)), \quad \bar{p}(t) = p_M \sum_{n=0}^{3} \exp\{\chi(t - (t_0 + n\, t_{\mathrm{per}}))^6\}, \tag{91}$$

which produce rapid small oscillations of the pressure, see Figure 5. We set the values of parameters $p_M = 4 \cdot 10^{-6}$, $\chi = -10^{15}$, $t_0 = 0.025$ and $t_{\mathrm{per}} = 0.05$. The calculation was carried out for $t \in (0, 0.2)$ with the aid of the 2-step ABDF- DGFE scheme with piecewise cubic approximation with respect to the space.

Figure 6 shows the isolines of pressure at several time instants. We observe a periodical propagation of pressure waves from the right to the left of the channel and also any reflection of these waves from the inflow which verifies the correctness of the presented boundary conditions.

It is also interesting to observe the size of time steps adaptively chosen by ABDF-DGFE method in a possible comparison by a fictitious explicit time discretization. So that we define the so-called cfl-value by

$$\mathrm{cfl}_k \equiv \tau_k \Lambda(\boldsymbol{w}_h^k), \quad k = 1, \ldots, r, \tag{92}$$

where $\Lambda(\cdot)$ is given by (29). It is a very well-known fact that, e.g., explicit first order time discretization of the finite volume space discretization (which is DGFE scheme with piecewise constant approximation) is stable if $\mathrm{cfl}_k < 1, \quad k = 1, \ldots, r$. For the higher degree of polynomial approximation with respect to space and explicit first order discretization with respect to time, we obtain still stronger restriction $\mathrm{cfl}_k \leq 0.2$, see [7, relation (70)].

Figure 7 shows the dependence of the cfl_k quantity on $t_k,\ k = 0, \ldots, r$. We observe that after several time steps at the beginning of the computation, we obtain (for t approximately greater than 0.01) a periodically repeating (with the period $\bar{T} = 0.05$) cfl-value. This value is (approximately) within the interval $(20, 45)$ when the pressure impulse from the outflow is propagated through the channel, whereas the cfl-value is up to 70 for time periods when the pressure impulses are out of the computational domain.

6.4 Forward facing step

The last example is the supersonic flow through the forward facing step, which represents a benchmark of the unsteady transonic inviscid compressible flow proposed by [25]. We employ the standard initial

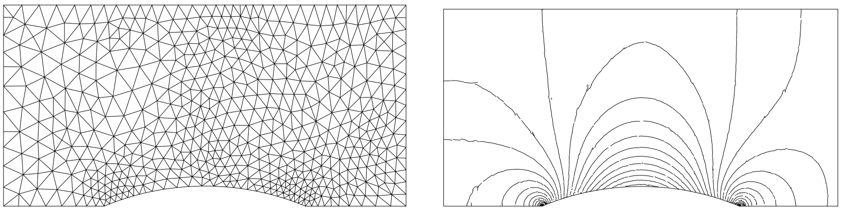


Fig. 4: GAMM channel, triangulation (left) and steady-state solution – isolines of pressure (right)

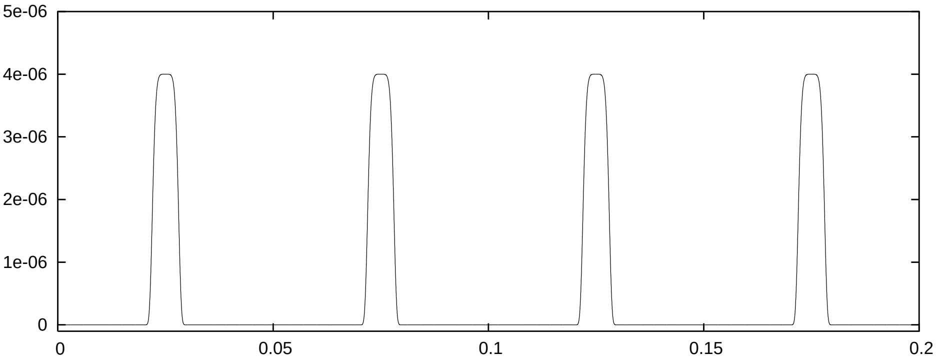


Fig. 5: Prescribed oscillation of pressure at outflow, function $\bar{p}(t)$ given by (91)

condition $\boldsymbol{w}_0 = (1.4,\ 3.,\ 0.,\ 1)$ which is also used as the boundary condition at the inflow/outflow parts of the step. We carried out the computation for the time interval $(0, 3)$ and employed an unstructured triangular grid having 13 371 elements. Figure 8 shows the isolines of density at $t = 0.5,\ 1.0,\ 1.5,\ 2.0,\ 2.5,\ 3.0$ obtained by the piecewise linear approximation. We observe the typical propagation of shock waves with multiple reflections. A good agreement with the reference solution from [25], pages 130-131, was achieved. In order to increase the sharpness of discontinuities, a suitable mesh adaptation strategy should be employed, which is a subject of further research. Figure 9 shows the dependence of the value cfl_k at time instants $t_k,\ k = 0, \dots, r$. The gain of the implicit time discretization is not high in comparison with the low Mach number example, since cfl_k is less than 10. So that the use of an explicit scheme should give the same results within a short computational time.

7 CONCLUSION

We carried out a discretization of the system of the Euler equations with the aid of the combination of the DGFE method for the space semi-discretization and the backward difference formula for the time discretization. We developed the adaptive strategy for the choice of the time step which we called the n-step ABDF-DGFE method ($n \geq 2$). It represents a higher order scheme with respect to the time and space coordinates. Several numerical examples verifying the efficiency of the algorithm were presented.

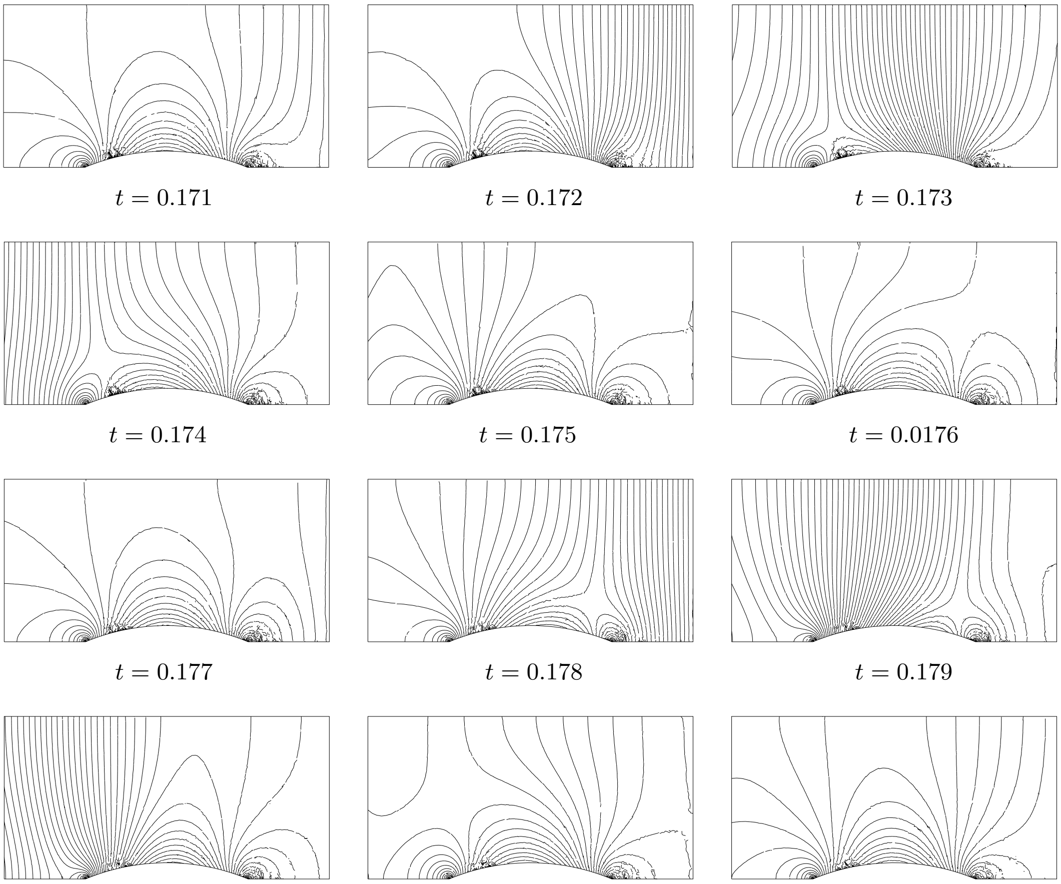
References

[1] Bassi F, Rebay S. High-order accurate discontinuous finite element solution of the 2D Euler equations. *J. Comput. Phys.*, 138:251–285, 1997.

[2] Cockburn B, Hou S, Shu CW. TVB Runge-Kutta local projection discontinuous Galerkin finite element for conservation laws IV: The multi-dimensional case. *Math. Comp.*, 54:545–581, 1990.

[3] Cockburn B, Karniadakis GE, Shu CV (editors). *Discontinuous Galerkin Methods.* Springer, Berlin, 2000.

[4] Crouzeix M. Une méthode multipsa implicit-explicit pour l'approximation des équations d'évolutions paraboliques. *Numer. Math.*, 35:27–276, 1980.



$t = 0.171$ $t = 0.172$ $t = 0.173$

$t = 0.174$ $t = 0.175$ $t = 0.0176$

$t = 0.177$ $t = 0.178$ $t = 0.179$

$t = 0.180$ $t = 0.181$ $t = 0.182$

Fig. 6: Low Mach number flow through the GAMM channel with oscillating outflow pressure, isolines of pressure at several time instants within one period

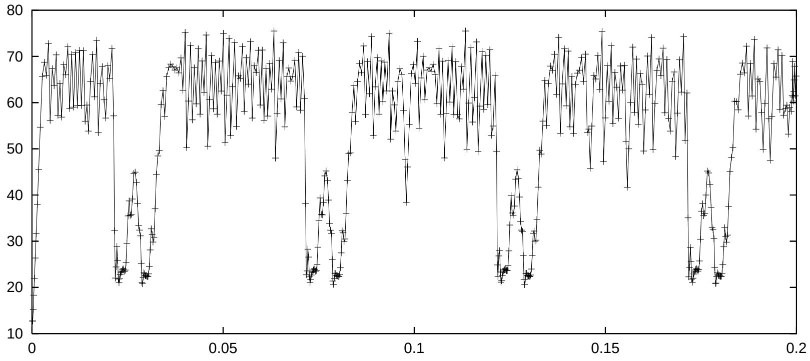
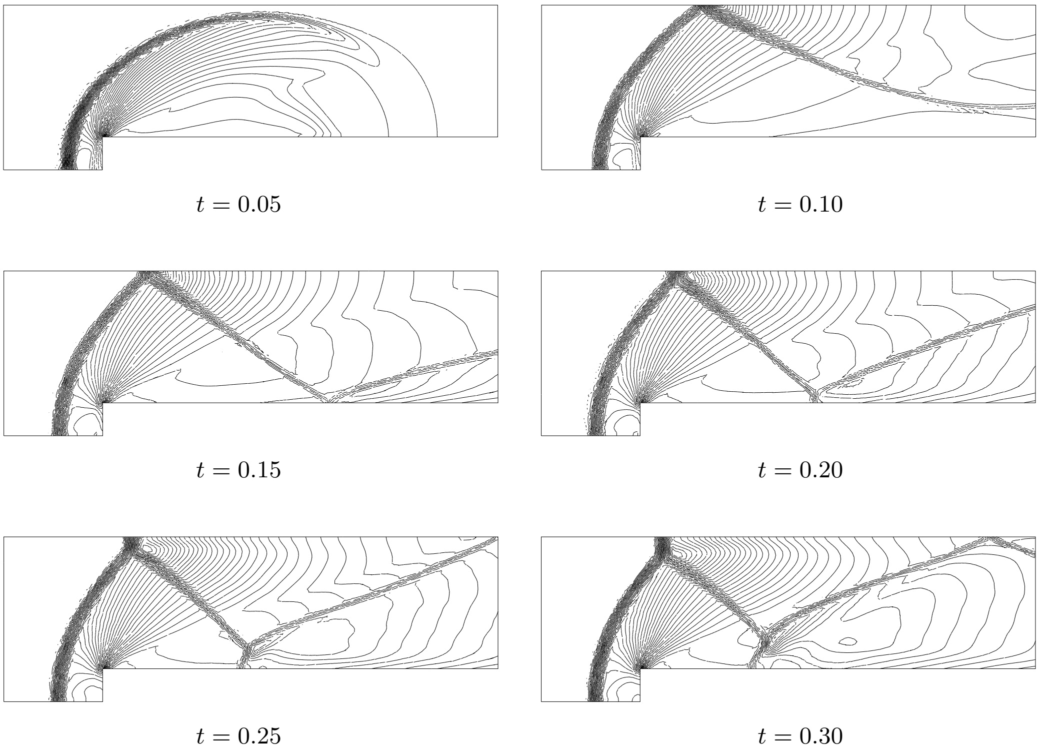


Fig. 7: GAMM channel, unsteady low Mach number flow, the dependence of cfl_k on $t_k,\ k = 0, \ldots, r$



$t = 0.05$

$t = 0.10$

$t = 0.15$

$t = 0.20$

$t = 0.25$

$t = 0.30$

Fig. 8: Unsteady flow though the forward facing step, isolines of density at several time instants

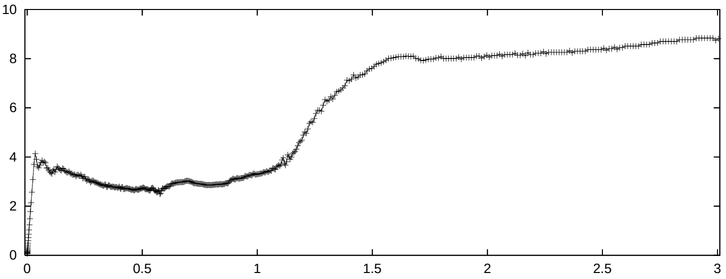


Fig. 9: Unsteady flow though the forward facing step, the dependence of cfl_k on $t_k,\ k = 0, \ldots, r$

[5] Dolejší V. Anisotropic mesh adaptation for finite volume and finite element methods on triangular meshes. *Comput. Vis. Sci.*, 1(3):165–178, 1998.

[6] Dolejší V. Anisotropic mesh adaptation technique for viscous flow simulation. *East-West J. Numer. Math.*, 9(1):1–24, 2001.

[7] Dolejší V. On the discontinuous Galerkin method for the numerical solution of the Navier–Stokes equations. *Int. J. Numer. Methods Fluids*, 45:1083–1106, 2004.

[8] Dolejší V. Higher order semi-implicit discontinuous Galerkin finite element schemes for compressible flow simulation. In I. Marek, editor, *Software and Algorithms of Numerical Mathematics, Srní, Bohemia*, pages 41–57, 2006.

[9] Dolejší V. Discontinuous Galerkin method for the numerical simulation of unsteady compressible flow. *WSEAS Transactions on Systems*, 5(5):1083–1090, 2006.

[10] Dolejší V. Higher order numerical schemes for hyperbolic systems with an application in fluid dynamics. In *Hyperbolic Probelms: Theory Numerics and Applications*, Eleventh international conference in Lyon, July 2006, (submitted).

[11] Dolejší V, Feistauer M. Semi-implicit discontinuous Galerkin finite element method for the numerical solution of inviscid compressible flow. *J. Comput. Phys.*, 198(2):727–746, 2004.

[12] Dolejší V, Feistauer M, Schwab C. On some aspects of the discontinuous Galerkin finite element method for conservation laws. *Math. Comput. Simul.*, 61:333–346, 2003.

[13] Feistauer M, Dolejší V, Kučera V. On a semi-implicit discontinuous Galerkin FEM for the nonstationary compressible Euler equations. In F. Asukara, H. Aiso, S. Kawashima, A. Matsumura, S. Nishibata, and K. Nishihara, editors, *Hyperbolic Probelms: Theory Numerics and Applications*, pages 391–398, Tenth international conference in Osaka, September 2004,, 2006. Yokohama Publishers, Inc.

[14] Feistauer M, Felcman J, Straškraba I. *Mathematical and Computational Methods for Compressible Flow.* Oxford University Press, Oxford, 2003.

[15] Feistauer M, Kučera V. On a robust discontinuous galerkin technique for the solution of compressible flow. *J. Comput. Phys.*, 2007. (in press).

[16] Hairer E, Norsett SP, Wanner G. *Solving ordinary differential equations I, Nonstiff problems.* Number 8 in Springer Series in Computational Mathematics. Springer Verlag, 2000.

[17] Hairer E, Wanner G. *Solving ordinary differential equations II, Stiff and differential-algebraic problems.* Springer Verlag, 2002.

[18] Hartmann R, Houston P. Adaptive discontinuous Galerkin finite element methods for the compressible Euler equations. *J. Comput. Phys.*, 183(2):508–532, 2002.

[19] Kůs P. Solution of convection-diffusion equations with the aid of higher order schemes with respect to space and time. Master's thesis, Charles University Prague, 2006.

[20] Lötstedt P, Söderberg S, Ramage A, Hemmingsson-Frändén L. Implicit solution of hyperbolic equations with space-time adaptivity. *BIT*, 42(1):134–158, 2002.

[21] van der Vegt JJW, van der Ven H. Space-time discontinuous Galerkin finite element method with dynamic grid motion for inviscid compressible flows. I: General formulation. *J. Comput. Phys.*, 182(2):546–585, 2002.

[22] Varah JM. Stability restrictions on second order, three level finite difference scheme for parabolic equations. *SIAM J. Numer. Anal.*, 17(2):300–309, 1980.

[23] Vijayasundaram G. Transonic flow simulation using upstream centered scheme of Godunov type in finite elements. *J. Comput. Phys.*, 63:416–433, 1986.

[24] Wesseling P. *Principles of Computational Fluid Dynamics.* Springer, Berlin, 2001.

[25] Woodward P, Colella P. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. of Comput. Phys.*, 54:115–173, 1984.